

Lesson 3: Drawing in Game Lab

45 minutes

Overview

This lesson is designed to give students a chance to get used to the programming environment, as well as the basic sequencing and debugging that they will use throughout the unit. Students begin with an introduction to the GameLab interactive development environment (IDE), then learn the three commands (`rect` , `ellipse` , and `fill`) that they will need to code the same types of images that they created on paper in the previous lesson. Challenge levels provide a chance for students who have more programming experience to further explore Game Lab.

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Standards

Full Course Alignment

CSTA K-12 Computer Science Standards (2017)

- **AP** - Algorithms & Programming

Agenda

Warm Up (5 minutes)

Programming Images

Activity (35 minutes)

Simple Drawing in Game Lab

Share Drawings

Wrap Up (5 minutes)

Objectives

Students will be able to:

- Sequence code correctly to overlay shapes.
- Use a coordinate system to place elements on the screen.

Preparation

- Prepare projector or other means of showing videos if you wish to watch as a class
- Check the "[Teacher's Lounge](#)" forum for verified teachers to find additional strategies or resources shared by fellow teachers
- If you are teaching virtually, consider checking our [Virtual Lesson Modifications](#)

Links

Heads Up! Please make a copy of any documents you plan to share with students.

For the teachers

- [Drawing Shapes](#) - Resource
- [Drawing in Game Lab](#) - Slides

▼ Make a Copy

For the students

- [Drawing in Game Lab - Part 1](#) - Video ([Download](#))
- [Drawing in Game Lab - Part 2](#) - Video ([Download](#))

Vocabulary

- **Bug** - Part of a program that does not work correctly.
- **Debugging** - Finding and fixing problems in an algorithm or program.
- **Program** - An algorithm that has been coded into something that can be run by a machine.

Introduced Code

- `ellipse(x, y, w, h)`
- `fill(color)`
- `rect(x, y, w, h)`

Teaching Guide

Warm Up (5 minutes)

Programming Images

Prompt: Based on what you know about computers, what do you think will be different between telling a person about your image and telling a computer about your image?

Share: Allow students time to think individually and discuss with a partner, then bring the class together and write their ideas on the board.

Discussion Goal: Guide students to realize that the way we communicate with each other may be more informal than how we communicate with a computer. For example, communicating with a digital assistant requires precise language and commands, usually starting with "Hey" and then the name of the device. Similarly, as we start coding in this unit, students will need to use a precise set of commands to communicate with a computer.

Remarks

In order to give instructions to a computer, we need to use a language that a computer understands. To make our animations and games, we will use a version of Javascript that uses blocks. The environment that we'll be programming in is called Game Lab.

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Activity (35 minutes)

Simple Drawing in Game Lab

Group: Place students in pairs to program together.

Pair programming is a great way to increase student confidence and foster the practices of collaboration and communication. You can read more about Pair Programming in the [CSD Guide to Teaching and Learning Strategies](#)

Transition: Send students to Code Studio.



1

Predict - Introduction to Game Lab

Teaching Tip

Facilitating Predict Levels: Predict levels are a great opportunity for students to think critically about code and engage in class discussions. Consider having students think of their own prediction and discuss with a partner before typing in their response. Once they've run the code, bring the class together for a full-group discussion to discuss how their predictions were similar or different from the resulting program. Use this as an opportunity to address any misconceptions that students may have had about the code initially.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

Discussion Goal: The parameters in this level will set the x and y values for the square. Students should recall this from the previous lesson, but what they may not realize is that the top left corner of the square is what will be placed at the 100 x and 100 y locations. Using this new information, they will be able to revisit this code at the next level and play with the x and y parameters to gain a better understanding.

Video: Show students the [Drawing in Game Lab - Part 1](#) video in the slides.

Teaching Tip

Videos are used throughout the curriculum to spark discussions, supplement key concepts with additional explanations and examples, and expose students to the various roles and backgrounds of individuals in computer science.

While interacting with the video, turn on closed captioning so students can also read along as they watch.

To encourage active engagement and reflection, use one or more of the strategies discussed in the [Guide to Curriculum Videos](#).

Questions to Consider with Video:

- Where can you find more information about how to use the blocks?
- What's an advantage of using block mode?

Discussion Goal:

- *Where can you find more information about how to use the blocks?*
 - Make sure students understand how to access the block documentation by clicking on the blocks inside the toolbox and clicking "see examples".
- *What's an advantage of using block mode?*
 - As students think of ideas of why they might prefer to use block mode, make sure that they understand that the block-based version of the programming language is just as legitimate as the text-based version. Students may offer that blocks make it easier to remember the exact commands or that they don't have to worry about the details of the parentheses or semicolons.
 - *Alternatively, advantages to text might be that it's easier to edit or that the text takes up less space.*

Using Resources: Below you can find recommendations for using the many resources students are introduced to in the lesson. You could consider creating a "Resource Chart" to keep track of these options and support students to be self-sufficient as they progress through levels.

- **Videos:** Watched as a class, but students can always return to them.
- **Help and Tips Tab:** This tab contains all of the relevant videos and reference guides for a particular level.
- **Reference Guides:** Contain text and diagrams explaining content. These are intended as helpful student resources, not class readings. They are a good place to go for review after learning content or when students get stuck in levels. You may decide to print these and have them available for students as they work through levels.
- **Documentation and Examples:** Hovering over a block will show a short description of what the block does. Clicking the "See Examples" link will open the documentation for that block.
- **Level Instructions:** Instructions may introduce small pieces of new content. Each level features a "Do This" section explaining what students are supposed to do in that level. Set the expectation early that reading these instructions, not just the "Do This" section, is important.



2-3

Skill Building

2

3

Facilitating Skill Building Levels: Skill Building levels are designed to continue teaching new skills and blocks through exploration, trial-and-error, and using worked examples from pre-supplied code. Students are still getting familiar with the concepts in the lesson and will need strong support throughout these levels to build confidence, debug their code, and cement their understanding.

Consider having students complete Skill Building levels in pairs using **Pair Programming**, which has students use one computer and trade between being a Driver or a Navigator. This process is highlighted in [this video](#), which you can show to the class. You can have students switch roles based on a timer, or switch every time they complete a level.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

Video: Show students the [Drawing in Game Lab - Part 2](#) video in the slides. **Questions to Consider with Video:**

- What's the difference between stroke and fill?

Discussion Goal: Stroke controls the border color of the shape, and fill controls the color inside of it.



4-6

Skill Building

4

5

6

Text-to-Speech Options: The instructions panel includes two options that can support comprehension for students.

- **Text to Speech** which reads aloud the instructions for students
- **Microsoft Immersive Reader** which opens a new panel for the instructions and gives controls to change the text size, contrast, or translate to another language.

[Click here to learn more about these options](#)



Practice

Teaching Tip

Facilitating Practice Levels: Practice levels are designed for students to apply their knowledge from the previous levels and develop fluency in using the new blocks of code to solve problems. Students can choose which practice levels they would like to complete, and it's not necessary for a student to complete each practice level before continuing.

Students tend to be more engaged and respond better when they have an authentic choice about how to continue their learning. Allow students to choose practice levels according to their interests and level of comfort, and consider providing opportunities for students to demonstrate and explain their solutions to the practice levels they chose to the entire class.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.



Assessment

Teaching Tip

Facilitating Assessment Levels: Assessment levels contain a single task that requires applying the skills and concepts from the level in order to solve. Students should complete these levels individually and you can use your judgment of how much external help students should have. Assessment levels also contain a rubric that can be used for formative assessment and a box to provide feedback to students - [click here to learn more about using rubrics and giving feedback to students](#).

Digging Deeper: For more tips about assessing student work, see the [CSD Guide to Assessment](#).

Assessment Opportunity

Formative Assessment: This level can be used as a formative assessment. A rubric is provided in the level, and written feedback can be given to students. [Click here to learn more about giving feedback to students](#).



Challenges

Teaching Tip

Facilitating Challenge Levels: Challenge levels are designed as extensions to the concepts and skills students learn throughout a lesson. Challenge levels tend to focus on more open-ended tasks for students to complete, or opportunities to combine several skills from previous lessons together into one program.

Challenge levels do **not** need to be completed for students to meet the core objectives of a lesson. Instead, every task in a challenge level is meant to supplement and enrich the learning objectives of a lesson, but are not required for future lessons. Students can still demonstrate mastery of the objectives of a lesson without completing any of the challenge levels.

Digging Deeper: For more tips about programming levels, see the [CSD Guide to Programming Levels](#). This document includes strategies and best-practices for facilitating programming levels with students.

Share Drawings

Share: Once students have completed their drawings, have them share them with the class. One way to do this is with a gallery walk.

Wrap Up (5 minutes)

Question of the Day: How can we communicate to a computer how to draw shapes on the screen?

Prompt: Today you learned how to draw in Game Lab for the first time. What type of advice would you share with a friend who was going to learn about drawing in Game Lab to make it easier for them?

Share: Allow students to share out their responses with the class.



This work is available under a [Creative Commons License \(CC BY-NC-SA 4.0\)](#).

If you are interested in licensing Code.org materials for commercial purposes [contact us](#).